Primal-Dual Interior Point Methods for Linear Programs

CS 520: Computational Methods in Optimization
Prof. David Gleich

Ishaan Saxena
isaxena@purdue.edu

In this project, I will study and implement a solver for Linear Programs in Julia using primal-dual interior point methods. In particular, I will look at the behavior of some path-following primal-dual methods, and will implement one or two such algorithms. This report will describe the relevant details of the project, the implementation, and the theory behind primal-dual interior point method which is utilized in the project.

1 Introduction

Consider a linear programming problem in the standard form. The primal and dual formulations for such a problem are given as follows:

$$\mathcal{P} = \begin{cases} \underset{\mathbf{x}}{\text{minimize}} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & \boldsymbol{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq 0 \end{cases} \tag{1.1a}$$

$$\mathcal{D} = \begin{cases} \underset{\boldsymbol{\lambda},\mathbf{s}}{\text{maximize}} & \mathbf{b}^T\boldsymbol{\lambda} \\ \text{subject to} & \boldsymbol{A}^T\boldsymbol{\lambda} + \mathbf{s} = \mathbf{c} \\ & \mathbf{s} \geq 0 \end{cases} \tag{1.1b}$$

where $\mathbf{c}, \mathbf{x}, \mathbf{s} \in \mathbb{R}^n$, $\boldsymbol{\lambda}, \mathbf{b} \in \mathbb{R}^m$, and $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ is a matrix with full row rank. It is easy to show that the solution to such a problem is characterized by the Karush-Kuhn-Tucker (KKT) conditions (Nocedal & Wright, 2006, Chapter 13), which are:

$$\boldsymbol{A}^T\boldsymbol{\lambda} + \mathbf{s} = \mathbf{c}, \tag{1.2a}$$
$$\boldsymbol{A}\mathbf{x} - \mathbf{b} = 0, \tag{1.2b}$$
$$\mathbf{x}^T\mathbf{s} = 0, \tag{1.2c}$$
$$\mathbf{x}, \mathbf{s} \geq 0 \tag{1.2d}$$

Primal-dual methods find the solution $(\mathbf{x}^*, \boldsymbol{\lambda}^*, \mathbf{s}^*)$ iteratively by applying variants of Newton's method to the three equality conditions as mentioned above to find search directions and use step lengths such that the inequality constraint is satisfied strictly in every iteration (Wright, 1997, Chapter 1).

1.1 Optimality Conditions as Non-Linear Equations

The KKT conditions can be restated into the following set of non-linear equations:

$$F(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{s}) = \begin{bmatrix} \boldsymbol{A}^T\boldsymbol{\lambda} + \mathbf{s} - c \\ \boldsymbol{A}\mathbf{x} - \mathbf{b} \\ \boldsymbol{X}\boldsymbol{S}\mathbf{e} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_c \\ \mathbf{r}_b \\ \mathbf{r}_{xs} \end{bmatrix} = 0 \tag{1.3a}$$

$$(\mathbf{x}, \mathbf{s}) \geq 0 \tag{1.3b}$$

where $\boldsymbol{X} = \text{diag}(\mathbf{x})$, $\boldsymbol{S} = \text{diag}(\mathbf{s})$, and $\mathbf{e}$ is a vector of all ones. To solve such a system, we can use Newton's Method. As a result, we obtain the following system of linear equations:

$$J(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{s}) \begin{bmatrix} \mathbf{p}_x \\ \mathbf{p}_\lambda \\ \mathbf{p}_s \end{bmatrix} = -F(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{s}) \tag{1.4}$$

where J is the Jacobian of F. Assuming our current point is strictly feasible, this gives us:

$$\begin{bmatrix} 0 & \boldsymbol{A}^T & \boldsymbol{I} \\ \boldsymbol{A} & 0 & 0 \\ \boldsymbol{S} & 0 & \boldsymbol{X} \end{bmatrix} \begin{bmatrix} \mathbf{p}_x \\ \mathbf{p}_\lambda \\ \mathbf{p}_s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\mathbf{r}_{xs} \end{bmatrix} \tag{1.5}$$

Solving this system gives us the search direction $(\mathbf{p}_x, \mathbf{p}_\lambda, \mathbf{p}_s)$. As a result, we obtain the following iterate:

$$(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{s}) + \alpha(\mathbf{p}_x, \mathbf{p}_\lambda, \mathbf{p}_s)$$

For some line search step length $\alpha \in (0, 1]$.

1.2 Drawbacks of Pure Newton's Method and Modifications

Recall the iterate we obtain with the basic Newton's method:

$$(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{s}) + \alpha(\mathbf{p}_x, \mathbf{p}_\lambda, \mathbf{p}_s)$$

The new iterate obtained from this method satisfies the constraints given in (1.2a), (1.2b) for any values of α. This is easily verifiable with some basic calculuations. However, it is the condition $(\mathbf{x}, \mathbf{s}) \geq 0$, which is slightly more challenging to deal with. Note that this condition implies that $\mathbf{x}^T\mathbf{s} \geq 0$. As a result, we obtain on the new iterate:

$$\begin{aligned} (\mathbf{x} + \alpha\mathbf{p}_x)^T(\mathbf{s} + \alpha\mathbf{p}_s) &\geq 0 \\ \mathbf{x}^T\mathbf{s} + \alpha(\mathbf{p}_x^T\mathbf{s} + \mathbf{p}_s^T\mathbf{x}) + \alpha^2\mathbf{p}_x^T\mathbf{p}_s &\geq 0 \\ (1 - \alpha)\mathbf{x}^T\mathbf{s} + \alpha^2\mathbf{p}_x^T\mathbf{p}_s &\geq 0 \end{aligned}$$

As such, the value of α is bound by this inequality. It turns out that the pure Newton's method allows only a very small step before the strict inequality constraint is violated, hence, we do not make much progress towards a solution (Wright, 1997, Chapter 1).

To fix this problem, primal-dual interior point methods make modifications to the basic Newton method in two important ways (Wright, 1997, Chapter 1):

1. They bias the search direction towards the interior of the negative orthant $(\mathbf{x}, \mathbf{s}) \geq 0$ such that we can move further along the search direction before the non-negativity constraint is violated.
2. They keep the components of $(\mathbf{x}, \mathbf{s})$ from moving too close to the boundaries of the feasible polytope without making much progress towards the solution.

2 Log-Barrier Terms to Modify LPs

Consider the following modification to our standard linear program:

$$\text{minimize } \mathbf{c}^T\mathbf{x} + b(\mathbf{x};\tau) \quad \text{subject to } \boldsymbol{A}\mathbf{x} = \mathbf{b} \tag{2.1}$$

where $b(\mathbf{x};\tau) = -\tau \sum_i \log x_i$.

Note that as $x_i \to 0$, we have $-\log(x_i) \to \infty$. As a result, given some positive τ, the function $b(\mathbf{x};\tau)$ creates a "barrier" for the optimization problem. If we get closer to zero, the value of $\mathbf{c}^T\mathbf{x} + b(\mathbf{x};\tau)$ spikes. As a result, to find a minimizer, we must ensure that $x_i > 0$, satisfying the initial non-negativity constraint.

Furthermore, note that we have $b(\mathbf{x};\tau) \to 0$ as $\tau \to 0$. This indicates that, the smaller the value of τ gets, the closer $\mathbf{c}^T\mathbf{x} + b(\mathbf{x};\tau)$ gets to $\mathbf{c}^T\mathbf{x}$. This means that at small values of τ, the log-barrier function b can generate approximations of the solution to the linear program, with smaller values of τ giving better approximations.

It is also important to note that $\mathbf{c}^T\mathbf{x}+b(\mathbf{x};\tau)$ is strictly convex, which indiciates that the optimality conditions are both necessary and sufficient. To find these optimality conditions, let's find the Lagrangian.

The Lagrangian for this optimization problem is:

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{c}^T\mathbf{x} - \tau \sum_i \log x_i - \boldsymbol{\lambda}^T(\boldsymbol{A}\mathbf{x} - \mathbf{b}) \tag{2.2}$$

As such, the optimality conditions are:

$$\mathbf{c} - \tau \boldsymbol{X}^{-1}\mathbf{e} - \boldsymbol{A}^T\boldsymbol{\lambda} = 0 \tag{2.3a}$$

$$\boldsymbol{A}\mathbf{x} - \mathbf{b} = 0 \tag{2.3b}$$

Since the non-negativity condition will be satisfied given this formulation of a linear program, we can obtain the KKT conditions in (1.2) by enforcing:

$$\tau \boldsymbol{X}^{-1} = \boldsymbol{S} \tag{2.4}$$

with the only difference being that here we require a strict inequality instead for $(\mathbf{x}, \mathbf{s}) > 0$, and that $x_i s_i = \tau$, not 0.

3 The Central Path

The primal-dual feasible set $\mathcal{F}$ and strictly feasible set $\mathcal{F}^0$ are defined as follows:

$$\mathcal{F} = \{(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{s}) | \boldsymbol{A}\mathbf{x} = \mathbf{b}, \boldsymbol{A}^T\boldsymbol{\lambda} + \mathbf{s} = \mathbf{c}, (\mathbf{x}, \mathbf{s}) \geq 0\} \tag{3.1a}$$

$$\mathcal{F}^0 = \{(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{s}) | \boldsymbol{A}\mathbf{x} = \mathbf{b}, \boldsymbol{A}^T\boldsymbol{\lambda} + \mathbf{s} = \mathbf{c}, (\mathbf{x}, \mathbf{s}) > 0\} \tag{3.1b}$$

The central path arc $\mathcal{C}$ is an arc of strictly feasible points parameterized by the scalar $\tau > 0$ such that each point $(\mathbf{x}_\tau, \boldsymbol{\lambda}_\tau, \mathbf{s}_\tau) \in \mathcal{C}$ satisfies the optimality conditions for the reformulation of the linear program with a log-barrier term. That is:

$$\boldsymbol{A}^T\boldsymbol{\lambda} + \mathbf{s} = \mathbf{c}, \tag{3.2a}$$

$$\boldsymbol{A}\mathbf{x} - \mathbf{b} = 0, \tag{3.2b}$$

$$x_i s_i = \tau, \tag{3.2c}$$

$$\mathbf{x}, \mathbf{s} > 0 \tag{3.2d}$$

As a result, we can define $\mathcal{C}$ with the mapping $F : \mathbb{R}^{2n+m} \to F : \mathbb{R}^{2n+m}$ defined in (1.3a) as follows:

$$F(\mathbf{x}_\tau, \boldsymbol{\lambda}_\tau, \mathbf{s}_\tau) = \begin{bmatrix} 0 \\ 0 \\ \tau\mathbf{e} \end{bmatrix}, \quad (\mathbf{x}_\tau, \mathbf{s}_\tau) > 0 \tag{3.3}$$

It can also be shown that $(\mathbf{x}_\tau, \mathbf{s}_\tau)$ is unique for each value of $\tau > 0$ if the linear program is feasible ($\mathcal{F}^0$ is nonempty) (Wright, 1997, Chapter 2). This indicates that as $\tau \to 0$, $\mathcal{C}$ converges to a primal-dual solution of the linear program. Thus, the central path guides us to the solution while maintaining $(\mathbf{x}, \mathbf{s}) > 0$ and decreasing $x_i s_i = \tau$ at the same rate.

As a result, we can modify our primal-dual interior point algorithm to take Newton steps towards points on $\mathcal{C}$ where $\tau > 0$ instead of taking pure Newton steps. Since these steps are biased towards the nonnegative orthant $(\mathbf{x}, \mathbf{s}) \geq 0$, we can take longer steps and make more significant progress before we would violate the positivity constraint.

3.1 Modified Search Directions

To describe this biased search directions, we introduce the following two terms:

1. the *centereing parameter* $\sigma \in [0, 1]$,
2. the *duality measure* μ:

$$\mu = \frac{1}{n}\sum_i x_i s_i = \frac{1}{n}\mathbf{x}^T\mathbf{s} = \frac{\tau}{n} \tag{3.4}$$

We use the following equations given these parameters to find the search direction for our current iterate (assuming the current point is feasible):

$$\begin{bmatrix} 0 & \boldsymbol{A}^T & \boldsymbol{I} \\ \boldsymbol{A} & 0 & 0 \\ \boldsymbol{S} & 0 & \boldsymbol{X} \end{bmatrix} \begin{bmatrix} \mathbf{p}_x \\ \mathbf{p}_\lambda \\ \mathbf{p}_s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\mathbf{r}_{xs} + \sigma\mu\mathbf{e} \end{bmatrix} \tag{3.5}$$

With these equations, setting $\sigma = 1$ defines a centering direction, in which the Newton step is towards a point in the central path with the same duality measure. On the other hand, $\sigma = 0$, we obtain the affine-scaling direction, in which the Newton step is the one which reduces the duality measure the most. It is important to note for the duality measure that:

$$\mu = \frac{1}{n}\mathbf{x}^T\mathbf{s} = \frac{1}{n}(\mathbf{c}^T\mathbf{x} - \mathbf{b}^T\boldsymbol{\lambda}) \tag{3.6}$$

As such, the more the duality measure is reduced, the closer we are to the optimum solution. Hence, in order to obtain the solution and make significant progress towards it at each step, we must make a trade off between reducing μ (smaller σ value) and staying close to the central path (larger σ value). This choice of σ value is, as a result, very important to the primal-dual interior point algorithm.

3.2 A Basic Framework for Primal-Dual Interior Point Methods

With the information we have obtained regarding the modified search direction with respect to the central path, we can obtain the following general framework for primal-dual interior point algorithms (Wright, 1997, Chapter 1):

Algorithm 1 Primal-Dual Interior Point Framework

Given: $(\mathbf{x}_0, \boldsymbol{\lambda}_0, \mathbf{s}_0) \in \mathcal{F}$
for $k = 0, 1, 2...$ **do**
  choose centering parameter $\sigma_k \in [0, 1]$
  solve equations in (3.5) to obtain search directions $\mathbf{p}_x, \mathbf{p}_\lambda, \mathbf{p}_s$
  choose step length α_k
  $(\mathbf{x}_{k+1}, \boldsymbol{\lambda}_{k+1}, \mathbf{s}_{k+1}) \leftarrow (\mathbf{x}_k, \boldsymbol{\lambda}_k, \mathbf{s}_k) + \alpha_k(\mathbf{p}_x, \mathbf{p}_\lambda, \mathbf{p}_s)$
end for

4 Path-Following Methods

Path-following methods[1] follow the central path $\mathcal{C}$ in the direction of decreasing τ in order to reach the solution set for the primal-dual problem. While the iterates do not stay exactly on $\mathcal{C}$, they are explicitly restricted to a well-defined neighborhood of it. Each search direction is, as previously discussed, a Newton step towards a point on $\mathcal{C}$, for which the duality measure τ is less than or same as the current duality measure μ. The target value $\tau = \sigma\mu$ is used where $\sigma \in [0, 1]$ is the centering parameter (Wright, 1997, Chapter 5).

Before we discuss some path-following methods in depth, let's introduce the notion of central path neighborhoods.

[1]For the rest of this project, I will restrict my focus to only path-following methods. While potential-reduction algorithms are an interesting topic for delibration, they utilize the central path neighborhoods only implicitly; I seem to be more interested in the workings and nature of path following methods.

4.1 Central Path Neighborhoods

Recall the optimality conditions for the log-barrier formulation of the stanard form linear program as described in equations (3.2) which define points on the central path parameterized by τ. Path-following algorithms generate iterates $(\mathbf{x}_k, \boldsymbol{\lambda}_k, \mathbf{s}_k) \in \mathcal{F}^0$. However, these deviate from points on the central path $\mathcal{C}$ since the pairwise products $x_i s_i$ do not generally match. As such, they satisfy all conditions in (3.2) except (3.2c).

In order to prevent premature jumps towards the boundary of the nonnegative orthant, we can define a measure for such deviation by comparing the pairwaise products to the duality measure μ (which refers to the average value of the pairwise products) with the following scaled norm (Wright, 1997, Chapter 5):

$$\frac{1}{\mu}||\boldsymbol{X}\boldsymbol{S}\mathbf{e} - \mu\mathbf{e}||_p \tag{4.1}$$

Now, suppose that the i^{th} component of either $\mathbf{x}$ or $\mathbf{s}$ is zero. This indicates that:

$$\frac{1}{\mu}||\boldsymbol{X}\boldsymbol{S}\mathbf{e} - \mu\mathbf{e}||_p \geq \frac{1}{\mu}|x_i s_i - \mu| = 1 \tag{4.2}$$

Thus, to ensure strict positivity of both $\mathbf{x}$ and $\mathbf{s}$, we can restrict the scaled norm deviation from a point on the central path to a strict upper bound of 1. This gives us the p-Neighborhood of $\mathcal{C}$:

$$\mathcal{N}_p(\theta) = \{(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{s}) \in \mathcal{F}^0 \mid ||\boldsymbol{X}\boldsymbol{S}\mathbf{e} - \mu\mathbf{e}||_p \leq \theta\mu\} \tag{4.3}$$

where $\theta \in [0, 1)$. The most commonly utilized of such neighborhoods is the $\mathcal{N}_2$ neighborhood with θ typically set to 0.5 (Nocedal & Wright, 2006, Chapter 14).

However, $\mathcal{N}_2$ can prove to be very restrictive in its nature. Note that for $\mathbf{x}, \boldsymbol{\lambda}, \mathbf{s} \in \mathcal{N}_2(\theta)$:

$$\left\{\sum_i (x_i s_i - \mu)^2 \leq \theta^2\mu^2\right\} \Rightarrow \left\{\sum_i \left(\frac{x_i s_i}{\mu} - 1\right)^2 \leq \theta^2 < 1\right\} \tag{4.4}$$

This indicates that the sum of all squared relative deviations cannot exceed 1 even if θ is set close to its upper bound. As a result, such a neighborhood may capture only a very small portion of the strictly feasible set $\mathcal{F}^0$. To allow for a more expansive neighborhood, in addition to neighborhoods based on the norm deviations, we can also define the $\mathcal{N}_{-\infty}$ neighborhood as follows (Wright, 1997, Chapter 5):

$$\mathcal{N}_{-\infty}(\gamma) = \{(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{s}) \in \mathcal{F}^0 \mid x_i s_i \geq \gamma\mu \ \forall i\} \tag{4.5}$$

where $\gamma \in (0, 1]$. If a point lies in this neighborhood, each pairwise product $x_i s_i$ must be at least some small multiple of the duality measure, thus, preventing premature jumps towards the boundary of the nonnegative orthant.

The path-following methods we will now discuss follow the framework described in Section 3.2. These methods choose either $\mathcal{N}_2$ or $\mathcal{N}_{-\infty}$ and then choose the centering parameter σ_k and step length α_k such that each iterate stays within the chosen neighborhood. By keeping all iteratres within the same neighborhood, such methods prevent them from coming too close to the boundary of the nonnegative orthant, therby ensuring nontrivial steps along the search direction to reduce all pairwise products at a similar rate.

4.2 Path-Following Algorithms Based on Central Path Neighborhoods

4.2.1 Short-Step Path-Following Algorithm

The short-step path-following algorithm, as described by Monteiro and Adler in their 1989 paper, is a variant of the basic primal-dual path-following paradigm we have discuessed. Given some fixed values for θ, δ satisfying

$$0 \leq \theta < \frac{1}{2}, \quad 0 < \delta < \sqrt{n}, \tag{4.6a}$$

$$\frac{\theta^2 + \delta^2}{2(1-\theta)} \leq \theta \left(1 - \frac{\delta}{\sqrt{n}}\right), \tag{4.6b}$$

where n is the number of columns in the constraint matrix $\boldsymbol{A}$, this algorithm starts at a point $(\mathbf{x}_0, \boldsymbol{\lambda}_0, \mathbf{s}_0) \in \mathcal{N}_2(\theta)$, and uses uniform values $\alpha = 1$ and $\sigma = 1 - \delta/\sqrt{n}$ for each iterate.

With this procedure for choosing the centering parameter σ_k and step length α_k, each iterate $(\mathbf{x}_k, \boldsymbol{\lambda}_k, \mathbf{s}_k)$ stays within the chosen neighborhood of the central path, and the duality measure μ_k converges linearly to zero at a rate $1 - \sigma$ (Monteiro & Adler, 1989, Sections 3 and 4).

This algorithm is as follows:

Algorithm 2 Short-Step Path-Following Algorithm

Given: some θ, δ satisfying (4.6) and $(\mathbf{x}_0, \boldsymbol{\lambda}_0, \mathbf{s}_0) \in \mathcal{N}_2(\theta)$
set centering parameter $\sigma = 1 - \delta/\sqrt{n}$
for $k = 0, 1, 2...$ **do**
  solve equations in (3.5) to obtain search directions $\mathbf{p}_x, \mathbf{p}_\lambda, \mathbf{p}_s$
  $(\mathbf{x}_{k+1}, \boldsymbol{\lambda}_{k+1}, \mathbf{s}_{k+1}) \leftarrow (\mathbf{x}_k, \boldsymbol{\lambda}_k, \mathbf{s}_k) + (\mathbf{p}_x, \mathbf{p}_\lambda, \mathbf{p}_s)$
end for

If the starting point $\mu_0 \in \mathcal{N}_2(\theta)$ such that, for some $\epsilon > 0$, $\mu_0 \leq 1/\epsilon^\kappa$ (given some positive constant κ), then this algorithm takes $K = O(\sqrt{n} \log 1/\epsilon)$ iterations to reach $\mu_k \leq \epsilon$ for all $k \geq K$ (Wright, 1997, Theorem 5.2).

This algorithm is globally convergent (Wright, 1997, Lemma 5.1). However, the duality measure converges to zero only at a linear rate. Furthermore, this algorithm uses a constant value for the centering parameter and step length at every iteration, and thereby lacks adaptivity in choice of these parameters.

4.2.2 Predictor-Corrector Algorithm

The Predictor-Corrector Algorithm (Mizuno, Todd, & Ye, 1993, Section 3) is a variant of the primal-dual path-following methods wherein it takes the following alternating steps:

1. predictor step ($\sigma = 0$) to reduce the duality measure,
2. corrector step ($\sigma = 1$) to improve centrality.

It utilizes a pair of nested neighborhoods $\mathcal{N}_2(0.25)$ and $\mathcal{N}_2(0.5)$ such that, given a starting point in the inner neighborhood, each iterate generated by the corrector step is restricted to the first, and each iterate generated by the predictor step is restricted to the second. As a result, the predictor step reduced the duality measure μ by a factor of $(1-\alpha)$, where α is the step length, whereas the corrector step merely brings the iterate back to a point on the central path $\mathcal{C}$ with the same μ (Wright, 1997, Chapter 5).

This algorithm is as follows:

Algorithm 3 Predictor-Corrector Algorithm

Given: $(\mathbf{x}_0, \boldsymbol{\lambda}_0, \mathbf{s}_0) \in \mathcal{N}_2(0.25)$
for $k = 0, 1, 2...$ **do**
  if k is even **then** ▷ predictor step
    $\sigma_k \leftarrow 0$
    solve equations in (3.5) to obtain search directions $\mathbf{p}_x, \mathbf{p}_\lambda, \mathbf{p}_s$
    choose $\alpha_k \in (0, 1]$ such that:

$$\begin{aligned} &\text{maximize} && \alpha && (4.7a)\\ &\text{subject to} && (\mathbf{x}_k(\alpha), \boldsymbol{\lambda}_k(\alpha), \mathbf{s}_k(\alpha)) \in \mathcal{N}_2(0.5) && (4.7b) \end{aligned}$$

    $(\mathbf{x}_{k+1}, \boldsymbol{\lambda}_{k+1}, \mathbf{s}_{k+1}) \leftarrow (\mathbf{x}_k, \boldsymbol{\lambda}_k, \mathbf{s}_k) + \alpha(\mathbf{p}_x, \mathbf{p}_\lambda, \mathbf{p}_s)$
  else ▷ corrector step
    $\sigma_k \leftarrow 1$
    solve equations in (3.5) to obtain search directions $\mathbf{p}_x, \mathbf{p}_\lambda, \mathbf{p}_s$
    $(\mathbf{x}_{k+1}, \boldsymbol{\lambda}_{k+1}, \mathbf{s}_{k+1}) \leftarrow (\mathbf{x}_k, \boldsymbol{\lambda}_k, \mathbf{s}_k) + (\mathbf{p}_x, \mathbf{p}_\lambda, \mathbf{p}_s)$
  end if
end for

Furthermore, it can be shown that equations (4.7) have a satisfactory analytic bound. Suppose the current iterate $(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{s}) \in \mathcal{N}_2(0.25)$. If choose α which satisfies

$$\overline{\alpha} = \min\left\{\frac{1}{2}, \left(\frac{\mu}{8||\boldsymbol{P}_x\boldsymbol{P}_s\mathbf{e}||}\right)^{1/2}\right\} \tag{4.8}$$

where $\boldsymbol{P}_x = \text{diag}(\mathbf{p}_x)$ and $\boldsymbol{P}_s = \text{diag}(\mathbf{p}_s)$, then we can ensure that $(\mathbf{x}_k(\alpha), \boldsymbol{\lambda}_k(\alpha), \mathbf{s}_k(\alpha)) \in \mathcal{N}_2(0.5)$ for all $\alpha \in [0, \overline{\alpha}]$ (Mizuno et al., 1993, Lemma 4). We can go one extra step to show that (Wright, 1997, Lemma 5.4):

$$\frac{\mu}{8||\boldsymbol{P}_x\boldsymbol{P}_s\mathbf{e}||} \geq \frac{0.16}{n} \Rightarrow \overline{\alpha} \geq \frac{0.4}{\sqrt{n}} \tag{4.9}$$

If the starting point $\mu_0 \in \mathcal{N}_2(0.25)$ such that, for some $\epsilon > 0$, $\mu_0 \leq 1/\epsilon^\kappa$ (given some positive constant κ), then this algorithm takes $K = O(\sqrt{n}\log 1/\epsilon)$ iterations to reach $\mu_k \leq \epsilon$ for all $k \geq K$ (Wright, 1997, Theorem 5.9).

This algorithm is globally convergent and does so at a superlinear rate (Wright, 1997, Theorem 7.7). It is a definite improvement over the short-step path-following method, insofar as it is more

adaptive due to its built in choice of σ at each step. This allows it to take larger steps in order to reduce μ in the predictor cycles, and maintain centering at the corrector cycles.

Despite its adaptivity, however, it is still restricted to the $\mathcal{N}_2$ neighborhood of the central path. This means that it cannot make significant progress during the initial steps of the algorithm when we are quite far from a solution.

4.2.3 Long-Step Path-Following Algorithm

Instead of using an $\mathcal{N}_2$ neighborhood of the central path, long-step path following methods generate a sequence of iterates in the neighborhood $\mathcal{N}_{-\infty}(\gamma)$, which, for values of γ very close to zero, occupies most of the strictly feasible set of points $\mathcal{F}^0$. Such an algorithm uses fixed values $\sigma_{\min}, \sigma_{\max}$ and chooses $\sigma_k \in [\sigma_{\min}, \sigma_{\max}]$. The step length α is chosen as the maximum value in $[0, 1]$ such that the next iterate remains in the chosen neighborhood.

This algorithm is as follows:

Algorithm 4 Long-Step Path-Following Algorithm

Given: $\gamma \in (0, 1), 0 < \sigma_{\min} < \sigma_{\max} < 1$, and some $(\mathbf{x}_0, \boldsymbol{\lambda}_0, \mathbf{s}_0) \in \mathcal{N}_{-\infty}(\gamma)$
for $k = 0, 1, 2...$ **do**
  choose $\sigma_k \in [\sigma_{\min}, \sigma_{\max}]$
  solve equations in (3.5) to obtain search directions $\mathbf{p}_x, \mathbf{p}_\lambda, \mathbf{p}_s$
  choose $\alpha_k \in (0, 1]$ such that:

$$\begin{aligned} &\text{maximize} && \alpha && \text{(4.10a)} \\ &\text{subject to} && (\mathbf{x}_k(\alpha), \boldsymbol{\lambda}_k(\alpha), \mathbf{s}_k(\alpha)) \in \mathcal{N}_{-\infty}(\gamma) && \text{(4.10b)} \end{aligned}$$

  $(\mathbf{x}_{k+1}, \boldsymbol{\lambda}_{k+1}, \mathbf{s}_{k+1}) \leftarrow (\mathbf{x}_k, \boldsymbol{\lambda}_k, \mathbf{s}_k) + \alpha(\mathbf{p}_x, \mathbf{p}_\lambda, \mathbf{p}_s)$
end for

Again, we can derive satisfactory analytic bounds for α using equations (4.10). It can be shown that if the current iterate $(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{s}) \in \mathcal{N}_{-\infty}(\gamma)$, then for all α such that

$$\alpha \in \left[0, 2^{3/2}\gamma \frac{1-\gamma}{1+\gamma}\frac{\sigma_k}{n}\right] \tag{4.11}$$

we obtain $(\mathbf{x}_k(\alpha), \boldsymbol{\lambda}_k(\alpha), \mathbf{s}_k(\alpha)) \in \mathcal{N}_{-\infty}(\gamma)$ (Wright, 1997, Theorem 5.11).

If the starting point $\mu_0 \in \mathcal{N}_{-\infty}(\gamma)$ such that, for some $\epsilon > 0$, $\mu_0 \leq 1/\epsilon^\kappa$ (given some positive constant κ), then this algorithm takes $K = O(n \log 1/\epsilon)$ iterations to reach $\mu_k \leq \epsilon$ for all $k \geq K$ (Wright, 1997, Theorem 5.12).

The bounds on σ in this algorithm ensure that each search direction moves off the boundary of the chosen neighborhood into its interior. Thus, small steps along the search direction improve centrality, but larger steps eventually move outside the neighborhood. The bounds on α in (4.11) guarantee that a certain minimum step length before which we reach the boundary of the neighborhood. However, while this algorithm is more closely related to practical implementations, the complexity bound is worse.

4.3 Convergence of Path-Following Algorithms

For each of the algorithms we have discussed, we have shown that given some $\epsilon > 0$, if the initial value of the duality measure is in the chosen neighborhood, that is, $\mu_0 \in \mathcal{N}_p(\theta)$, then there exists some positive integer K such that $\mu_k \leq \epsilon$ for all $k \geq K$. This is the same as saying:

$$\mu_k \to 0 \tag{4.12}$$

Consider some subsequence $\mathcal{K}$ of $(\mathbf{x}_k, \boldsymbol{\lambda}_k, \mathbf{s}_k)$, such that:

$$\lim_{k \in \mathcal{K}} (\mathbf{x}_k, \boldsymbol{\lambda}_k, \mathbf{s}_k) = (\mathbf{x}, \boldsymbol{\lambda}, \mathbf{s}) \tag{4.13}$$

Since we restrict each term in $\mathcal{K}$ to a chosen neighborhood, we know for all $k \in \mathcal{K}$ that:

$$(\mathbf{x}_k, \boldsymbol{\lambda}_k, \mathbf{s}_k) \in \mathcal{F}^0$$

This indicates that we obtain a subsequence of iterates all of which satisfy:

$$\boldsymbol{A}\mathbf{x}_k = \mathbf{b}, \quad \boldsymbol{A}^T\boldsymbol{\lambda}_k + \mathbf{s}_k = \mathbf{c}, \quad (\mathbf{x}_k, \mathbf{s}_k) > 0 \tag{4.14}$$

Note that since $\mu_k = \mathbf{x}^T\mathbf{s}/n$, we can conclude that at a limit point of $\mathcal{K}$, we would arrive at the following conditions:

$$\boldsymbol{A}\mathbf{x} = \mathbf{b}, \quad \boldsymbol{A}^T\boldsymbol{\lambda} + \mathbf{s} = \mathbf{c}, \quad (\mathbf{x}, \mathbf{s}) > 0, \quad \mathbf{x}^T\mathbf{s} = 0 \tag{4.15}$$

Since these match the KKT conditions as defined in (1.2), we can prove that the path-following interior-point algorithms we have discussed in section 4.2 all converge to the solution of the primal-dual linear program.

5 Infeasible Starting Points

The algorithms we have discussed so far require that we have feasible starting points. Choice of the starting point is a very important issue as it has effects on the robustness of the algorithm. However, it is often hard to find a point which is contained within the strictly feasible set $\mathcal{F}^0$; there exist some valid linear program where this set is empty (Wright, 1997, Chapter 6). While there are several methods to overcome such a problem by reformulation of the linear program, etc., we will now discuss variants of the Long-Step Path-Following Algorithm (Section 4.2.3) and the Predictor-Corrector algorithm (Section 4.2.2) which do not require a strictly feasible starting point.

5.1 Infeasible-Path-Following Algorithm

This algorithm is very similar to algorithms used in practice. The steps in this algorithm are obtained by solving the following variant of the Newton equations discussed in (3.5):

$$\begin{bmatrix} 0 & \boldsymbol{A}^T & \boldsymbol{I} \\ \boldsymbol{A} & 0 & 0 \\ \boldsymbol{S} & 0 & \boldsymbol{X} \end{bmatrix} \begin{bmatrix} \mathbf{p}_x \\ \mathbf{p}_\lambda \\ \mathbf{p}_s \end{bmatrix} = \begin{bmatrix} -\mathbf{r}_c \\ -\mathbf{r}_b \\ -\mathbf{r}_{xs} + \sigma\mu\mathbf{e} \end{bmatrix} \tag{5.1}$$

where, as usual, $\mathbf{r}_c = \boldsymbol{A}^T\boldsymbol{\lambda} + \mathbf{s} - c$, $\mathbf{r}_b = \boldsymbol{A}\mathbf{x} - \mathbf{b}$, and $\mathbf{r}_{xs} = \boldsymbol{X}\boldsymbol{S}\mathbf{e}$.

This algorithm restricts iterates to an extension of the $\mathcal{N}_{-\infty}(\gamma)$ such that each iterate generated is infeasible and the limit is both feasible and optimal. This extended neighborhood is defined as follows:

$$\mathcal{N}_{-\infty}(\gamma,\beta) = \left\{ (\mathbf{x},\boldsymbol{\lambda},\mathbf{s}) \;\middle|\; ||(\mathbf{r}_b,\mathbf{r}_c)|| \le \beta\mu\frac{||(\mathbf{r}_b^{(0)},\mathbf{r}_c^{(0)})||}{\mu_0},\ (\mathbf{x},\mathbf{s}) > 0,\ \mathbf{x}_i\mathbf{s}_i \ge \gamma\mu\ \forall i \right\} \tag{5.2}$$

where $\gamma \in (0,1)$ and $\beta \ge 1$. As a result, the infeasibility of all points in this neighborhood is uniformly bound by the duality measure μ. By forcing $\mu_k \to 0$, we ensure that $\mathbf{r}_b \to 0$ and $\mathbf{r}_c \to 0$ as $k \to \infty$.

Furthermore, we impose new conditions on the choice of the step length α which guarantee that the amount by which constraints (3.2a) and (3.2b) are violated decrease at least as rapidly as the duality measure and that there is a sufficient decrease in the duality measure at each step.

This algorithm is as follows:

Algorithm 5 Infeasible-Path-Following Algorithm

Given: $\gamma \in (0,1), \beta \ge 1, 0 < \sigma_{\min} < \sigma_{\max} < 0.5$, and some $(\mathbf{x}_0,\boldsymbol{\lambda}_0,\mathbf{s}_0) \in \mathcal{N}_{-\infty}(\gamma,\beta)$
for $k = 0, 1, 2...$ **do**
  choose $\sigma_k \in [\sigma_{\min}, \sigma_{\max}]$
  solve equations in (5.1) to obtain search directions $\mathbf{p}_x, \mathbf{p}_\lambda, \mathbf{p}_s$
  choose $\alpha_k \in (0,1]$ such that:

$$\begin{aligned} &\text{maximize} && \alpha && (5.3a)\\ &\text{subject to} && (\mathbf{x}_k(\alpha),\boldsymbol{\lambda}_k(\alpha),\mathbf{s}_k(\alpha)) \in \mathcal{N}_{-\infty}(\gamma,\beta) && (5.3b)\\ &&& \mu_k(\alpha) \le (1-0.01\alpha)\mu_k && (5.3c) \end{aligned}$$

  $(\mathbf{x}_{k+1},\boldsymbol{\lambda}_{k+1},\mathbf{s}_{k+1}) \leftarrow (\mathbf{x}_k,\boldsymbol{\lambda}_k,\mathbf{s}_k) + \alpha(\mathbf{p}_x,\mathbf{p}_\lambda,\mathbf{p}_s)$
end for

When given a feasible starting point, this algorithm mimics the behavior of the long-step path-following algorithm as discussed in (4.2.3). This algorithm is globally convergent, wherein the duality gap μ_k converges Q-linearly to 0, and the residual measure $||(\mathbf{r}_b^{(k)},\mathbf{r}_c^{(k)})||$ converges R-linearly to 0 (Wright, 1997, Theorem 6.1).

With some fairly general conditions on the starting point, this algorithm takes $K = O(n^2|\log\epsilon|)$ iterations to reach $\mu_k \le \epsilon$ for all $k \ge K$ (Wright, 1997, Theorem 6.2).

While this algorithm gives a basic groundwork for convergence from an infeasible starting point, it does not converge fast enough. Nor does this algorithm have a good enough theoretical complexity bound.

5.2 Potra's Predictor-Corrector Method

Potra's algorithm, like the infeasible-path-following algorithm utilizes a neighborhood which allows infeasible points in order to solve the linear programming problem. For feasible starting points, this algorithm reduces to the Mizuno-Todd-Ye predictor-corrector algorithm discussed in (4.2.2) (Potra, 1994). The predictor steps in this algorithm are obtained by solving the variant of the Newton equations discussed in (5.1), whereas the corrector steps solve the Newton equations discussed previously in (3.5).

5.2.1 Algorithm

We start off with the assumption that the feasible solution set $\mathcal{F}$ (3.1a) is nonempty, i.e., some optimal primal-dual solution exists. In addition to this feasible solution set $\mathcal{F}$, we define the ϵ-approximate solution set $\mathcal{F}_\epsilon$ for some $\epsilon > 0$ as follows:

$$\mathcal{F}_\epsilon = \{(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{s}) \;\mid\; ||\boldsymbol{A}\mathbf{x} - \mathbf{b}|| \leq \epsilon, ||\boldsymbol{A}^T\boldsymbol{\lambda} + \mathbf{s} - \mathbf{c}|| \leq \epsilon, \mathbf{x}^T\mathbf{s} \leq \epsilon, (\mathbf{x}, \mathbf{s}) > 0\} \tag{5.4}$$

This algorithm uses a variant of the $\mathcal{N}_2(\theta)$ neighborhood defined as follows:

$$\mathcal{N}_\theta = \{(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{s}) \in \mathbb{R}^{2n+m} \mid ||\boldsymbol{X}\boldsymbol{S}\mathbf{e} - \mu\mathbf{e}|| \leq \theta\mu, (\mathbf{x}, \mathbf{s}) > 0\} \tag{5.5}$$

Thus, this algorithm restricts the norm of the element-wise difference from the duality measure by some multiple of the duality measure (Potra, 1994, Equation 6). Starting from some point $(\mathbf{x}_0, \boldsymbol{\lambda}_0, \mathbf{s}_0) \in \mathcal{N}_\theta$, this algorithm produces an ϵ-approximate solution.

Furthermore, this algorithm depends on the two parameters θ, β which satisfy:

$$0 < \frac{\beta^2}{2\sqrt{2}(1-\beta)} < \theta < \beta < 1 \tag{5.6}$$

These correspond to the neighborhood parameters for the predictor-corrector algorithm when the starting point is feasible (it is easy to show that $\theta = 0.25$ and $\beta = 0.5$ satisfy this requirement).

We also define the following measures at each step of the algorithm:

$$\mathbf{f} = \boldsymbol{X}_k\boldsymbol{S}_k\mathbf{e} - \mu_k\mathbf{e}, \quad \mathbf{g} = \boldsymbol{P}_x\boldsymbol{P}_s\mathbf{e} - \frac{\mathbf{p}_x^T\mathbf{p}_s}{n}\mathbf{e} \tag{5.7}$$

These are use to calculate the following modified values of the inner neighborhood parameter θ:

$$\theta_0 = \beta^2\mu_k^2 - ||\mathbf{f}||^2, \quad \theta_1 = \mathbf{f}^T\mathbf{g} - \frac{1}{n}\mathbf{p}_x^T\mathbf{p}_s\mu_k\beta^2, \quad \theta_2 = ||\mathbf{g}||^2 - \left(\frac{1}{n}\mathbf{p}_x^T\mathbf{p}_s\right)^2\beta^2 \tag{5.8}$$

We can define modified values of the outer neighborhood parameter as follows:

$$\beta_0 = \sqrt{8}\theta(1-\beta) - \beta^2, \quad \beta_1 = \frac{2\beta||\boldsymbol{P}_x\boldsymbol{P}_s\mathbf{e}||}{\mu_k} \tag{5.9}$$

The following values are utilized when computing the step length for the predictor step:

$$\alpha_1 = \frac{1}{2}\left(-\varphi_1 + \sqrt{\varphi_1^2 + 4\varphi_1}\right) \tag{5.10a}$$

$$\alpha_2 = \frac{1}{2}\left(-\varphi_2 + \sqrt{\varphi_2^2 + 4\varphi_2}\right) \tag{5.10b}$$

$$\alpha_3 = \frac{1}{2\beta_1}\left(\beta^2 + \sqrt{\beta^4 + 4\beta_0\beta_1}\right) \tag{5.10c}$$

where φ_1, φ_2 are defined as follows:

$$\varphi_1 = \theta_0/(\theta_1 + \sqrt{\theta_1^2 + \theta_0\theta_2}), \quad \varphi_2 = \mu_k\sqrt{\sqrt{8}(\beta-\theta)(1-\beta)}/||\boldsymbol{P}_x\boldsymbol{P}_s\mathbf{e}|| \tag{5.10d}$$

Algorithm 6 Potra's Predictor-Corrector Method

Given: $\mathbf{z}_0 = (\mathbf{x}_0, \boldsymbol{\lambda}_0, \mathbf{s}_0) \in \mathcal{N}_\theta$, $\psi_0 = 1$, parameters (θ, β)
for $k = 0, 1, 2...$ **do**
  solve equations in (5.1) with $\sigma = 0$ to obtain $\mathbf{p}_{\text{aff}} = (\mathbf{p}_x, \mathbf{p}_\lambda, \mathbf{p}_s)$ ▷ predictor step
  compute $\theta_0, \theta_1, \theta_2$ as in equation (5.8)
  if $(\theta_1^2 + \theta_0\theta_2 \leq 0)$ or $(\theta_1, \theta_2 \leq 0)$ **then**
    $\alpha_1 \leftarrow 1$
  else
    set α_1 as in equation (5.10a)
  end if
  if $\mathbf{p}_x^T\mathbf{p}_s = 0$ **then** ▷ orthogonal search directions
    $\overline{\alpha} \leftarrow \min\{1, \alpha_1\}$
    $\hat{\mathbf{z}} = \mathbf{z}_k + \overline{\alpha}\mathbf{p}_{\text{aff}}$
  else ▷ non-orthogonal search directions
    compute α_2, α_3 as in equation (5.10)
    $\overline{\alpha} \leftarrow \min\{1, \alpha_1, \alpha_2, \alpha_3\}$
    $\overline{\mathbf{z}} = \mathbf{z}_k + \overline{\alpha}\mathbf{p}_{\text{aff}}$
    solve equations the following linear system to correct for $\mathbf{p}_x^T\mathbf{p}_s \neq 0$ in the infeasibility case:

$$\begin{bmatrix} 0 & \boldsymbol{A}^T & \boldsymbol{I} \\ \boldsymbol{A} & 0 & 0 \\ \overline{\boldsymbol{S}} & 0 & \overline{\boldsymbol{X}} \end{bmatrix} \begin{bmatrix} \overline{\mathbf{p}}_x \\ \overline{\mathbf{p}}_\lambda \\ \overline{\mathbf{p}}_s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\boldsymbol{P}_x\boldsymbol{P}_s\mathbf{e} \end{bmatrix} \tag{5.11}$$

    to obtain $\overline{\mathbf{p}}_{\text{aff}} = (\overline{\mathbf{p}}_x, \overline{\mathbf{p}}_\lambda, \overline{\mathbf{p}}_s)$
    $\hat{\mathbf{z}} = \overline{\mathbf{z}} + \overline{\alpha}^2\overline{\mathbf{p}}_{\text{aff}}$
  end if
  solve equations in (3.5) with $\sigma = 1$ to obtain $\mathbf{p}_{\text{cor}} = (\mathbf{p}_x, \mathbf{p}_\lambda, \mathbf{p}_s)$ ▷ corrector step
  $\mathbf{z}_{k+1} \leftarrow \hat{\mathbf{z}} + \mathbf{p}_{\text{cor}}$
  $\psi_{k+1} \leftarrow (1 - \overline{\alpha})\psi_k$
end for

Here, we can see three notable steps. In addition to the predictor and corrector step, this algorithm solves the system described in 5.11 to account for $\mathbf{p}_x^T\mathbf{p}_s \neq 0$ in the case that the iterate is not

a feasible point. The predictor and corrector step, however, are the same as the Mizuno-Todd-Ye predictor-corrector algorithm, as described previously. As a result, when the starting point is feasible, this algorithm reduces to the Mizuno-Todd-Ye predictor-corrector algorithm, which uses $\theta = 0.25, \beta = 0.5$ (Potra, 1994, Page 388).

This algorithm is convergent globally and takes $K = O(n^{\omega} \log \epsilon_0/\epsilon)$ iterations to reach $\mu_k \leq \epsilon$ for all $k \geq K$ where $\omega = 1/2$ or 1 depending on the starting point and $\epsilon_0 = \max\{\mu_0, ||\mathbf{r}_b^{(0)}||, ||\mathbf{r}_c^{(0)}||\}$ (Potra, 1994, Section 3). Furthermore, this algorithm also converges quadratically from any starting point $(\mathbf{x}_0, \boldsymbol{\lambda}_0, \mathbf{s}_0) \in \mathcal{N}_\theta$ (Potra, 1994, Section 4).

5.2.2 Detecting Infeasibility with Potra's Algorithm

At the beginning of every iteration, we check for the following termination condition:

$$\mathbf{x}_k^T \mathbf{s}_k \leq \epsilon, \quad ||\mathbf{r}_b^{(k)}|| \leq \epsilon, \quad ||\mathbf{r}_c^{(k)}|| \leq \epsilon \tag{5.12}$$

if each of these conditions is fulfilled, we declare that our current iterate $\mathbf{z}_k$ is an ϵ-approximate solution, that is $\mathbf{z}_k \in \mathcal{F}_\epsilon$.

It can be shown that if $\mathcal{F}$ is nonempty then the sequence generated by Potra's algorithm satisfies the following property (Potra, 1994, Lemma 3.1):

$$\mathbf{x}_0^T \mathbf{s}_k + \mathbf{s}_0^T \mathbf{x}_k \leq n(\mu_k + \mu_0) + \left(1 - \frac{\mu_k}{\mu_0}\right)(\mathbf{x}_0^T \mathbf{s}^* + \mathbf{s}_0^T \mathbf{x}^*) \tag{5.13}$$

If the set $\mathcal{F}$ is nonempty then the algorithm terminates in a finite number of steps given any $\epsilon > 0$ with some $\mathbf{z}_k \in \mathcal{F}_\epsilon$ (Potra, 1994, Theorem 3.4).

As a result, if we have the following result at the beginning of an iteration:

$$\mathbf{x}_0^T \mathbf{s}_k + \mathbf{s}_0^T \mathbf{x}_k \geq n(\mu_k + \mu_0) + \left(1 - \frac{\mu_k}{\mu_0}\right)(\overline{\rho}_d ||\mathbf{x}_0|| + \overline{\rho}_p \mathbf{s}_0^T \mathbf{x}^*) \tag{5.14}$$

then there exist no $(\mathbf{x}^*, \boldsymbol{\lambda}^*, \mathbf{s}^*) \in \mathcal{F}$ for which $||\mathbf{x}^*|| \leq \overline{\rho}_d$ and $||\mathbf{s}^*|| \leq \overline{\rho}_p$. If the algorithm terminates, it does so either with the conditions in (5.12), in which case it is an ϵ-approximate solution or it does so eith the conditions in (5.14), in which case the problem is infeasible (Potra, 1994, Theorem 5.4).

5.3 Mehrotra's Predictor-Corrector Algorithm

While Potra's Predictor-Corrector Algorithm gives us plenty advantages, such as the ability to start from an infeasible point, superlinear convergence, and polynomial time complexity, it also has some drawbacks which make it less desirable in practice.

Potra's algorithm solves three different linear systems, each with a different coefficient matrix in each iteration. As a result, much of the time is spent on the linear algebra. This makes these algorithms much slower for practical use. However, such an algorithm lays impoartant groundwork for algorithms used in practice.

As we will discuss in the following section on solving the linear system, each instance wherein we need to solve a linear system of equations requires that we compute the Cholesky decomposition of the coefficient matrix. This primal-dual interior-point method, similar to Potra's method, uses this fact reuse the same factorization of a matrix multiple times in each iteration to yield information about the affine and centering steps (Mehrotra, 1992).

A key feature of this algorithm is the reuse of the Cholesky decomposition obtained for the coefficient matrix twice. First, we solve for an affine-scaling direction (equation (5.1) with $\sigma = 0$). Recall the three steps which Potra's method takes: this method combines the predictor, corrector, and centering steps into one. The initial affine-scaling directions are used to find the centering parameter σ in order to achieve this.

This algorithm is as follows:

Algorithm 7 Mehrotra's Predictor-Corrector Algorithm

Given: $(\mathbf{x}_0, \boldsymbol{\lambda}_0, \mathbf{s}_0)$

for $k = 0, 1, 2...$ **do**

solve equations in (5.1) to obtain search directions $\mathbf{p}_x^{\text{aff}}, \mathbf{p}_\lambda^{\text{aff}}, \mathbf{p}_s^{\text{aff}}$

Compute $\alpha_k^{\text{pri}}, \alpha_k^{\text{dual}} \in (0, 1]$ such that:

$$\alpha_k^{\text{pri}} = \min_{i:\mathbf{p}_{x_i}^{\text{aff}}<0} \left\{ -\frac{x_{k_i}}{\mathbf{p}_{x_i}} \right\} \quad \alpha_k^{\text{dual}} = \min_{i:\mathbf{p}_{s_i}^{\text{aff}}<0} \left\{ -\frac{s_{k_i}}{\mathbf{p}_{s_i}} \right\} \tag{5.15}$$

Set $\mu_{\text{aff}} = \frac{1}{n}\mathbf{x}(\alpha_k^{\text{pri}})^T\mathbf{s}(\alpha_k^{\text{dual}})$

Set $\sigma = (\mu_{\text{aff}}/\mu)^3$

Compute the search directions by solving the following linear system:

$$\begin{bmatrix} 0 & \boldsymbol{A}^T & \boldsymbol{I} \\ \boldsymbol{A} & 0 & 0 \\ \boldsymbol{S} & 0 & \boldsymbol{X} \end{bmatrix} \begin{bmatrix} \mathbf{p}_x \\ \mathbf{p}_\lambda \\ \mathbf{p}_s \end{bmatrix} = \begin{bmatrix} -\mathbf{r}_c \\ -\mathbf{r}_b \\ -\boldsymbol{X}\boldsymbol{S}\mathbf{e} - \boldsymbol{P}_x\boldsymbol{P}_s\mathbf{e} + \sigma\mu\mathbf{e} \end{bmatrix} \tag{5.16}$$

Recompute $\alpha_k^{\text{pri}}, \alpha_k^{\text{dual}} \in (0, 1]$ as in (6.1)

Choose $\eta \in (0.9, 1]$ and update:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \eta\alpha_k^{\text{pri}}\mathbf{p}_x \tag{5.17a}$$

$$[\boldsymbol{\lambda}_{k+1}, \mathbf{s}_{k+1}] = [\boldsymbol{\lambda}_k, \mathbf{s}_k] + \eta\alpha_k^{\text{dual}}[\mathbf{p}_\lambda, \mathbf{p}_s] \tag{5.17b}$$

end for

With this algorithm, however, there is no guarantee for convergence. However, divergent cases are rare and it converges fairly fast in other cases. Furthermore, it requires only one instance of matrix factorization for each iteration, and also has polynomial time complexity (Mehrotra, 1992; Nocedal & Wright, 2006). At the same time, we can also show that if a search direction computed in the last step rectifies the infeasibility in the residuals, we would obtain:

$$\mathbf{r}_b^{(k+1)} = (1 - \alpha_k^{\text{pri}})\mathbf{r}_b^{(k)}, \quad \mathbf{r}_c^{(k+1)} = (1 - \alpha_k^{\text{dual}})\mathbf{r}_c^{(k)} \tag{5.18}$$

This shows that there is, at least in some cases of the search direction and the points obtained in each iteration, a reasonable decrease in the infeasibility.

5.3.1 Choosing a Starting Point

Choosing a good starting point for an algorithm is a very important task. A good starting point leads to faster convergence for problems in general. We can modify the following popular heuristic (Wright, 1997, Chapter 11) slightly in order to fit the requirements for this algorithm and choose a point which satisfies:

Consider the following:

$$\text{minimize} \quad \frac{1}{2}\mathbf{x}^T\mathbf{x} \quad \text{subject to } \boldsymbol{A}\mathbf{x} = \mathbf{b} \tag{5.19a}$$

$$\text{minimize} \quad \frac{1}{2}\mathbf{s}^T\mathbf{s} \quad \text{subject to } \boldsymbol{A}^T\boldsymbol{\lambda} + \mathbf{s} = \mathbf{c} \tag{5.19b}$$

It can be easily shown that the solution for $(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{s})$ satisfying these equations can be written as follows (Nocedal & Wright, 2006, Page 410):

$$\widetilde{\mathbf{x}} = \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T)^{-1}\mathbf{b}, \quad \widetilde{\boldsymbol{\lambda}} = (\boldsymbol{A}\boldsymbol{A}^T)^{-1}\boldsymbol{A}\mathbf{c}, \quad \widetilde{\mathbf{s}} = \mathbf{c} - \boldsymbol{A}^T\boldsymbol{\lambda} \tag{5.20}$$

Now, let:

$$\delta_x = \max\{(-3/2)\min_i \widetilde{x}_i, 0\}, \quad \delta_s = \max\{(-3/2)\min_i \widetilde{s}_i, 0\} \tag{5.21}$$

We compute the following intermediate values:

$$\hat{\mathbf{x}} = \widetilde{\mathbf{x}} + \delta_x\mathbf{e}, \quad \hat{\mathbf{s}} = \widetilde{\mathbf{s}} + \delta_s\mathbf{e} \tag{5.22}$$

And obtain the starting values as follows:

$$\mathbf{x}_0 = \hat{\mathbf{x}} + \frac{1}{2}\frac{\hat{\mathbf{x}}^T\hat{\mathbf{s}}}{\mathbf{e}^T\mathbf{s}}\mathbf{e}, \quad \mathbf{s}_0 = \hat{\mathbf{s}} + \frac{1}{2}\frac{\hat{\mathbf{x}}^T\hat{\mathbf{s}}}{\mathbf{e}^T\mathbf{x}}\mathbf{e}, \quad \boldsymbol{\lambda}_0 = \widetilde{\boldsymbol{\lambda}} \tag{5.23}$$

With this, we can guarantee that the initial values of $\mathbf{x}, \mathbf{s}$ are not only nonnegative, but also somewhat close to eachother, and avoiding abnormally large values within their components. As a result, we can expect a uniform decrease in all components of the pairwise products.

6 Solving the Linear System

In any primal-dual algorithm, most of the computational effort is required in solving the linear systems which describe the search directions. In most applications, the matrix $\boldsymbol{A}$ is large and sparse. As such, the coefficient matrix (the Jacobian) is also large and sparse. The size of this matrix makes the computation much harder, and thus, this must be addressed, such that we can do this more efficiently.

Recall the linear system as discussed multiple times previously:

$$\begin{bmatrix} 0 & \boldsymbol{A}^T & \boldsymbol{I} \\ \boldsymbol{A} & 0 & 0 \\ \boldsymbol{S} & 0 & \boldsymbol{X} \end{bmatrix} \begin{bmatrix} \mathbf{p}_x \\ \mathbf{p}_\lambda \\ \mathbf{p}_s \end{bmatrix} = \begin{bmatrix} -\mathbf{r}_c \\ -\mathbf{r}_b \\ -\mathbf{r}_{xs} + \sigma\mu\mathbf{e} \end{bmatrix} \tag{6.1}$$

by removing $\mathbf{p}_s$ from this system, such that (since $\boldsymbol{X}, \boldsymbol{S}$ are diagonal and nonsingualar):

$$\begin{bmatrix} 0 & \boldsymbol{A} \\ \boldsymbol{A}^T & -\boldsymbol{D}^{-2} \end{bmatrix} \begin{bmatrix} \mathbf{p}_\lambda \\ \mathbf{p}_x \end{bmatrix} = \begin{bmatrix} -\mathbf{r}_b \\ -\mathbf{r}_c + \mathbf{s} - \sigma\mu\boldsymbol{X}^{-1}\mathbf{e} \end{bmatrix} \tag{6.2a}$$

$$\mathbf{p}_s = -\mathbf{s} + \sigma\mu\boldsymbol{X}^{-1}\mathbf{e} - \boldsymbol{X}^{-1}\boldsymbol{S}\mathbf{p}_x \tag{6.2b}$$

where we have introduced:

$$\boldsymbol{D} = \boldsymbol{S}^{-1/2}\boldsymbol{X}^{1/2} \tag{6.3}$$

This form of the step equations is known as the augmented system of step equations. Note that here $\boldsymbol{D}^{-2} = \boldsymbol{X}^{-1}\boldsymbol{S}$ is also diagonal and nonsingualar. As a result, can proceed to eliminate $\mathbf{p}_x$ from the step equations as well. This gives us the normal equations, which are as follows:

$$\boldsymbol{A}\boldsymbol{D}^2\boldsymbol{A}^T\mathbf{p}_\lambda = -\mathbf{r}_b + \boldsymbol{A}(-\boldsymbol{S}^{-1}\boldsymbol{X}\mathbf{r}_c + \mathbf{x} - \sigma\mu\boldsymbol{S}^{-1}\mathbf{e}) \tag{6.4a}$$

$$\mathbf{p}_s = -\mathbf{r}_c - \boldsymbol{A}^T\mathbf{p}_\lambda \tag{6.4b}$$

$$\mathbf{p}_x = -\mathbf{x} + \sigma\mu\boldsymbol{S}^{-1}\mathbf{e} - \boldsymbol{S}^{-1}\boldsymbol{X}\mathbf{p}_s \tag{6.4c}$$

Note that when $\mathbf{r}_b = 0$, this system reduces to the normal equations which represent the solution to the linear least squares problem with coefficient matrix $\boldsymbol{D}\boldsymbol{A}^T$. In order to obtain the step equations, we can apply sparse Cholesky decomposition on the matrix $\boldsymbol{A}\boldsymbol{D}^2\boldsymbol{A}^T$ and proceed. However, in the case that the matrix $\boldsymbol{A}\boldsymbol{D}^2\boldsymbol{A}^T$ is ill-conditioned and singular, we may use modified Cholesky methods to obtain a surrogate for the matrix $\boldsymbol{A}\boldsymbol{D}^2\boldsymbol{A}^T$ such that it differs only on the diagonal insofar as it makes the surrogate symmetrix positive definite (Fang & Oleary, 2008).

Let the $\boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^T$ modified Cholesky decomposition of the matrix $\boldsymbol{A}\boldsymbol{D}^2\boldsymbol{A}^T$ be:

$$\text{Modified-Cholesky}(\boldsymbol{A}\boldsymbol{D}^2\boldsymbol{A}^T) = \boldsymbol{L}\boldsymbol{M}\boldsymbol{L}^T \tag{6.5}$$

We know for such a decomposition that $\boldsymbol{L}\boldsymbol{M}\boldsymbol{L}^T = \boldsymbol{A}\boldsymbol{D}^2\boldsymbol{A}^T + \boldsymbol{E}$ such that (Fang & Oleary, 2008):

1. if $\boldsymbol{A}\boldsymbol{D}^2\boldsymbol{A}^T$ is sufficiently positive definite, $\boldsymbol{E}$ is very close to 0,
2. if $\boldsymbol{A}\boldsymbol{D}^2\boldsymbol{A}^T$ is not positive definite, $\boldsymbol{E}$ is not much larger that $\inf\{||\Delta\boldsymbol{A}||_p \mid \boldsymbol{A} + \Delta\boldsymbol{A} \succ 0\}$,
3. $\boldsymbol{A}\boldsymbol{D}^2\boldsymbol{A}^T + \boldsymbol{E}$ is reasonably well-conditioned.

Thus, in the case that the coefficient matrix $\boldsymbol{A}\boldsymbol{D}^2\boldsymbol{A}^T$ is well-conditioned, the modified Cholesky decomposition gives us this matrix, but when it is ill-conditioned, we obtain with the decomposition a surrogate which is positive definite.

We can utilize this to solve for $\mathbf{p}_\lambda$ in the following system:

$$(\boldsymbol{L}\boldsymbol{M}\boldsymbol{L}^T)\mathbf{p}_\lambda = \mathbf{v}_\lambda \tag{6.6}$$

where $\mathbf{v}_\lambda = -\mathbf{r}_b + \boldsymbol{A}(-\boldsymbol{S}^{-1}\boldsymbol{X}\mathbf{r}_c + \mathbf{x} - \sigma\mu\boldsymbol{S}^{-1}\mathbf{e})$. We can solve this system with the following steps:

1. solve the system $(\boldsymbol{L}\boldsymbol{M}^{1/2})\mathbf{y} = \mathbf{v}_\lambda$ to obtain $\mathbf{y}$,
2. solve the system $(\boldsymbol{L}\boldsymbol{M}^{1/2})^T\mathbf{p}_\lambda = \mathbf{y}$ to obtain $\mathbf{p}_\lambda$.

With this, we are solving a much smaller linear system than the one described in equations (6.1), and can do so with direct sparse Cholesky decomposition algorithms.

We can extend the same technique to solve similar linear systems as in equation (5.11). This gives us the following equations:

$$\boldsymbol{A}\boldsymbol{D}^2\boldsymbol{A}^T\overline{\mathbf{p}}_\lambda = -\boldsymbol{A}\overline{\boldsymbol{S}}^{-1}\boldsymbol{P}_x\boldsymbol{P}_s\mathbf{e} \tag{6.7a}$$

$$\overline{\mathbf{p}}_s = -\boldsymbol{A}^T\overline{\mathbf{p}}_\lambda \tag{6.7b}$$

$$\overline{\mathbf{p}}_x = -\overline{\boldsymbol{S}}^{-1}(\boldsymbol{P}_x\boldsymbol{P}_s\mathbf{e} + \overline{\boldsymbol{X}}\overline{\mathbf{p}}_s) \tag{6.7c}$$

Note that solving this linear system instead of one which has the form of (6.1) or (6.2) can only be useful when the matrix $\boldsymbol{A}$ is sufficiently sparse. A good metric to check if the normal equations are inappropriate is to check if $\boldsymbol{A}\boldsymbol{S}^{-1}\boldsymbol{X}\boldsymbol{A}^T$ is much denser than $\boldsymbol{A}$ (Wright, 1997, Page 17). In such a case, we can modify the normal equations appropriately.

7 Conclusions

Primal-dual interior point methods have several variants, each with their own advantages and disadvantages. These methods provide a fast, flexible way to deal with Linear Programs, due to the presence of their many formulations.

In this project, I primarily explored path-following primal-dual algorithms. First, I looked into some standard algorithms from which more practically feasible algorithms can be derived. I implemented Potra's predictor-corrector algorithm and Mehrotra's predictor-corrector algorithm over the course of this project. While Mehrotra's algorithm does not have proofs for convergence (indeed, in some cases, the algorithm diverges), such cases are rare in practice. The speed of the algorithm's convergence and the marginally low cost of linear algebra in this algorithm, due to its exploitation of the same matrix factorization multiple times in a single iteration make it a very desirable algorithm for most practical use.

Further, I briefly discussed the usage of such matrix factorizations (like the LDLt factorization) to more efficiently solve the linear systems which these methods used.

References

Fang, H.-r., & Oleary, D. P. (2008). Modified cholesky algorithms: a catalog with new approaches. *Mathematical Programming*, *115*(2), 319–349.

Mehrotra, S. (1992). On the implementation of a primal-dual interior point method. *SIAM Journal on optimization*, *2*(4), 575–601.

Mizuno, S., Todd, M. J., & Ye, Y. (1993). On adaptive-step primal-dual interior-point algorithms for linear programming. *Mathematics of Operations research*, *18*(4), 964–981.

Monteiro, R. D., & Adler, I. (1989). Interior path following primal-dual algorithms. part i: Linear programming. *Mathematical programming*, *44*(1-3), 27–41.

Nocedal, J., & Wright, S. (2006). *Numerical optimization*. Springer Science & Business Media.

Potra, F. A. (1994). A quadratically convergent predictorcorrector method for solving linear programs from infeasible starting points. *Mathematical Programming*, *67*(1-3), 383–406.

Wright, S. J. (1997). *Primal-dual interior-point methods* (Vol. 54). Siam.